Microsoft

Operating System

Deployment Guide:
Automating the Windows 2000 Upgrade

White Paper

Abstract

This deployment guide provides information and tips and tricks that will help you to automate the Microsoft® Windows® 2000 operating system upgrade process. It is designed for Information Systems professionals who are responsible for installing Windows 2000 Professional or one of the Windows 2000 Server Family products on many computers.

Microsoft Corporation • One Microsoft Way • Redmond, WA 98052-6399 • USA

0699

CONTENTS

INTRODUCTION

This deployment guide provides information and tips and tricks that will help you to automate the Microsoft® Windows® 2000 operating system upgrade process. It is designed for Information Systems professionals who are responsible for installing Windows 2000 Professional or one of the Windows 2000 Server Family products on many computers.

The term *upgrade* as it is used in this guide means replacing the operating system in place using the upgrade features of Windows 2000. A *migration* or *clean install* means formatting the disk and reinstalling the operating system and all applications.

Windows 9x is used throughout this guide to collectively refer to Microsoft Windows 95 and Microsoft Windows 98.

Most of the information presented is equally applicable to desktops and servers. Where there is a difference, specific reference will be made to the appropriate information. This guide also does not focus specifically on the upgrade to Windows 2000 Advanced Server or Windows 2000 Datacenter Server.

The Windows 2000 installation directory for Intel architecture computers is named i386. For Alpha architecture computers, this directory is named Alpha. Wherever reference is made to the i386 directory, the reader may substitute Alpha to refer to Alpha architecture computers.

This guide does not discuss “clean” installations on new computers, on freshly formatted disks, or in a new directory on an existing installation. Information on hard disk duplication, including cloning, can be found at http://www.microsoft.com/windows

In addition, this guide does not discuss the upgrade of server infrastructures or upgrading to the Windows 2000 Active Directory™ directory service.

You should use this guide in conjunction with the *Windows 2000 Planning and Design Guide* in the Windows 2000 Server Resource Kit.

WHAT YOU NEED TO KNOW

The first choice you will have to make is whether to upgrade existing installations or to perform a clean install of Windows 2000. You can upgrade from the following versions of the Microsoft Windows and Microsoft Windows NT® operating systems.

Supported Upgrade Paths

You can upgrade to Windows 2000 Professional from the following operating systems:

- Windows 95. All released versions including OSR2.x. See note 1.
- Windows 98
- Windows NT 3.51 Workstation
- Windows NT 4.0 Workstation
- Windows 2000 Professional. See note 2.

You can upgrade to Windows 2000 Server from the following operating systems:

- Windows NT 3.51 Server
- Windows NT 4.0 Server
- Windows NT 4.0 Terminal Server. See note 3.
- Windows 2000 Server. See note 2.

You can upgrade to Windows 2000 Advanced Server from all of the operating systems listed under the upgrade to Windows 2000 Server (above), as well as the following operating systems:

- Windows NT 4.0 Enterprise Edition
- Windows 2000 Advanced Server. See note 2.

Upgrades are not supported from the following operating systems:

- Windows 3.x, including Windows for Workgroups
- Versions of Windows NT prior to version 3.51
- BackOffice® Small Business Server
- Non Microsoft operating systems

Notes

1. Windows 95 supports network-based installations where the desktop operating system is shared from a server. Windows 2000 does not support this method of installation. If you are running network-based installations of Windows 95, you must perform a clean installation of Windows 2000 and reinstall the computer's applications.
2. Pre-release versions of Windows 2000 prior to Beta 3 do not support upgrades.
3. Upgrading from Windows NT 3.51 Server with Citrix is not supported.

Choosing Between an Upgrade and a Clean Installation

The decision on whether to upgrade or to perform a clean install is based on many factors, including.

- The level of configuration management and change control you are able to exert over your systems. If your environment is closely managed, with applications shared across the network and all user data centrally stored, then an upgrade could be the best option for you. If, however, your configuration state is not well controlled, a clean install could be the best option. Reformatting the hard disk and installing Windows 2000 and all of the applications results in a completely known configuration. Thereafter, you can use the improved manageability features of Windows 2000 and the Active Directory to closely control the configuration and ensure that a known state is maintained as new applications and updates are installed.

  Note If you upgraded (as opposed to performing a clean install) from Windows 3.x or Windows for Workgroups to Windows 9x, you should consider performing a clean installation of Windows 2000. It is likely that there are legacy issues associated with these upgrades that you could continue to inherit if you upgrade to Windows 2000. In this case, you might decide that a clean installation of Windows 2000 is preferred.

- User settings and application data. Even if all of a user's data and applications are stored centrally, there are user settings and application data stored on the local computer—for example, user preferences stored in the system registry. When performing a clean installation, these settings and data must be preserved and reapplied to the system after the installation. Not all of this data is easily identified, and extensive testing is required to ensure that you have identified everything that needs to be preserved. For example, many applications store data outside of the system registry in .ini files or proprietary configuration stores.

- The operating systems you have in your current environment: Windows 9x or a previous version of Windows NT. Upgrading from previous versions of Windows NT (version 3.51 and later) is inherently easier than upgrading from Windows 9x. This is due to the commonality between the operating system kernel architecture, device driver models, registry database, security architecture, and file systems. Upgrading from existing Windows 9x installations can present additional issues that need to be resolved.

- The mixture of desktop operating systems and types of network operating systems. It may well be the case that a mixture of upgrades and clean installs is appropriate within your environment. Much of the information presented in this guide is equally applicable to clean installations of Windows 2000 Professional and the Windows 2000 Server Family.

Push Compared to On-demand Installations

There are several different ways of executing Setup. In essence, these approaches can be grouped into two types of installation: *push* and *on-demand*. In general, a push install is one where Setup is executed on the target computer (known as the client) under the direction of a master computer, for example Microsoft Systems Management Server (SMS). An on-demand install is where the Setup command line is executed on the client computer in response to some action initiated at that client computer—for example from a logon script or a batch job on a network boot disk.

The common methods of executing Setup are as follows:

- By executing Winnt32.exe from a command line in Windows 9x and Windows NT.
- Using a network management installation package distributed by SMS or similar products.
- From a bootable CD-ROM mounted on the client computer.
- From a network boot disk, usually initiated by a support engineer.
- Through a shortcut or batch job attached to an email message.
- From a command issued in a user logon script.

The last two methods are not preferred as, in general, they do not provide any control of the day or time when Setup executes or the number of concurrent installations being performed. If you were to place a command line to upgrade to Windows 2000 in a user's logon script, when would that script execute? You may find that you have 300 users all logging in at more or less the same time—for example, first thing in the morning—thus causing a large demand for server resources and network bandwidth. The same is true of an object or shortcut attached to an email message. You have no control over when the upgrades are performed.

The preferred method of initiating the upgrade is to distribute the package with a network management software package or at the desktop with an upgrade initiated by a support engineer. By initiating the upgrade through an installation package distributed with SMS or other network management software, you can control how many concurrent upgrades are performed and when they occur. You can also take advantage of load balancing between multiple distribution servers and make optimal use of your available network bandwidth.

Having an engineer initiate the upgrade from a network boot disk or CD-ROM allows you to retain full control over the upgrade, and ensures that an engineer is available should anything go wrong. If you choose to use engineers, you should be sure that they are face to face with the customer so that they have a positive impact on user satisfaction. However, this is obviously more expensive than performing the upgrades centrally.

The creation and distribution of SMS packages is beyond the scope of this guide, but, in essence, SMS packages distribute a command line to client computers. The

issues of load balancing, timing of the execution of this command line, use of available bandwidth, and so on, are controlled by SMS. This guide focuses on how to create a Windows 2000 Setup distribution point and the associated answer file for unattended Setup. The command line used to initiate an unattended upgrade is thus common to all of the distribution mechanisms and to push and on-demand installs. Only the manner in which the Setup command line is executed varies.

Upgrading from Windows NT

While the upgrade from previous versions of Windows NT is easier than upgrading from Windows 9x, the following features and applications cannot be properly upgraded:

- Applications that depend on file system filters, for example anti-virus software and disk quota software. This is due to changes in the Windows Installable File System model. For detailed information, please refer to http://www.microsoft.com/hwdev/ntifskit/
- Networking protocols and clients that do not have updates on the Windows 2000 CD. Updated components can be found in the Winntupg folder of the i386 directory.
- Custom power management solutions and tools. Windows 2000 support for Advanced Configuration and Power Interface (ACPI) and Advanced Power Management (APM) replace these. You should remove the custom tools and solutions before upgrading.
- Custom Plug and Play solutions. These are no longer necessary as Windows 2000 provides full Plug and Play support. You should remove the custom solutions before upgrading.

Contact the software vendor to determine the availability of Windows 2000 compatible upgrades.

Upgrading from Windows 9x

The following Windows 9x features and applications cannot be upgraded:

- System utilities and features not supported in Windows 2000. For example, compressed drives that use DriveSpace or third-party applications and disk utilities, such as ScanDisk, disk defragmenters, and anti-virus programs. Compressed drives must be decompressed before upgrading.
- Applications and utilities that use virtual device drivers (VxDs) and .386 drivers. Check the [386Enh] section of the System.ini file to see if your system is loading any of these virtual drivers. Note that some device drivers make use of VxDs to provide property pages in property dialogs.
- Third-party Control Panel applications. These are often installed by device driver installation programs to provide additional functionality not provided out of the box with Windows 9x—for example, device-specific capabilities for display adapter drivers. You should test Control Panel application functionality as part of your Windows 2000 evaluation.

- Network components that do not ship on the Windows 2000 compact disc. Some of these components may have updates in the WIN9XUPG folder of the i386 directory.
- Custom power management solutions and tools. Windows 2000 ACPI and APM support replaces these. You should remove the custom tools and solutions before upgrading.
- Custom Plug and Play solutions. These are no longer necessary as Windows 2000 provides full Plug and Play support. You should remove the custom solutions before upgrading.

If the functionality provided by these features is not replaced by Windows 2000 and is still required, contact the software vendor to determine the availability of Windows 2000–compatible versions.

Because of the way that application developers design their Setup routines, many applications install differently on Windows 9x than on Windows 2000. For example, applications can:

- Maintain registry data in different locations between Windows 9x and Windows 2000 or Windows NT
- Make calls to Windows 9x–specific application programming interfaces
- Install different files when installed on Windows 9x than when installed on Windows 2000 or Windows NT

To address these issues, software vendors and corporate developers can implement upgrade packs that can move registry keys, install new versions of files, or move files within the file system. These upgrade packs are used during Windows 2000 Setup to resolve these incompatibilities. The upgrade pack contains a migration dynamic-link library (DLL) that Setup calls to update the application installation. The migration DLL mechanism is fully extensible. More information for developers is available in the latest Windows Platform SDK or at http://msdn.microsoft.com/developer/windows2000/default.asp

Because Windows 9x upgrades can require more planning and testing than upgrades from Windows NT, Windows 2000 Setup provides a “report only” mode that can generate compatibility reports and store them in a central location. You can then analyze these reports to determine whether you need migration DLLs and/or new versions of applications.

The upgrade process from Windows 9x is discussed in more detail later in this guide.

How Setup Detects Incompatibilities

Detecting Incompatibilities in Windows NT 3.51 and 4.0

You can use the Dosnet.inf file to control the detection of incompatible components and the way in which Setup continues if it finds them. Setup processes the Dosnet.inf file at the beginning of the Setup operation. You can configure the file to instruct Windows 2000 Setup to either cancel the upgrade or to allow the upgrade to

proceed if incompatibilities are encountered. If the incompatible component is a service, Setup can be instructed to continue and disable the incompatible service.

You can modify this file to include known incompatible components not detailed in Windows 2000. Details of these features of Dosnet.inf are provided later in this guide.

Detecting Incompatibilities in Windows 9x

The way in which Setup detects incompatibilities in Windows 9x is more complex. For hardware, Setup compares all Plug and Play IDs in the Windows 9x registry against a database of the Plug and Play devices supported on the Windows 2000 CD. If a device is not supported and it is not the device controlling the boot device, a wizard page appears prompting the user to provide Windows 2000 drivers for these devices. Setup also checks entries in the computer's Config.sys file to determine if there are any 16-bit device drivers referenced. If any are found, a warning is added to the compatibility report.

For software, Setup scans the computer and determines whether any programs use Windows 9x–specific APIs that are not available in Windows 2000. It also compares all of the executable files against a database of known issues. When Setup finds an incompatibility, an entry is written to the compatibility report in the following categories:

- Programs that do not work under Windows 2000. These programs must be upgraded or replaced. Upgrade packs might be supplied for these applications.
- Programs that work but with minor issues. Some programs work but exhibit minor problems; for example, the Microsoft Magic School Bus Human Body may exhibit display problems showing full screen animations.
- Programs that must be removed before upgrading. Some programs affect Windows 2000 Setup and can prevent a successful upgrade. These applications must be removed before upgrading.
- Programs that are removed by Setup. Some programs are no longer required due to additional functionality in Windows 2000. For example, Adobe Type Fonts are now managed through the Control Panel Fonts application. The Adobe Type Manager Control Panel application is therefore removed during Setup.
- Programs that should be reinstalled. Some programs may install in a different manner under Windows 9x than under Windows 2000. These programs will work when reinstalled after Windows 2000 Setup has completed. For example, to run correctly, Microsoft Close Combat 1.0 must be reinstalled after Windows 2000 Setup. This is because there is no upgrade pack for Microsoft Close Combat.

These problems are reported when you run Windows 2000 Setup in the check upgrade only mode. See the Release Notes for Windows 2000 for details of applications that can prevent the upgrade from being successful.

Checking Computers for Compatibility

During the early stages of upgrade planning, it is vital that you conduct an audit of your software and hardware for compatibility with Windows 2000. This audit is used to identify applications and hardware that need to be upgrade or replaced. Systems management software, such as Microsoft Systems Management Server, often provides this functionality. In addition, Windows 2000 Setup provides a report only mode that can be used to start the compatibility checking process.

Checking Windows 9x Systems

When run under Windows 9x, Windows 2000 Setup can save the generated report to a central location. To do this, use the following command line.

```
Winnt32.exe /unattend:answer_file
```

where *answer_file* is a fully qualified file name of a minimal Windows 2000 Setup answer file, as follows:

```
[Unattended]
Win9xUpgrade=yes

[Win9xUpg]
ReportOnly=Yes
SaveReportTo=report_file
```

report_file is the fully qualified file name of the report file to generate. Note that you can use %computername% anywhere in the path. Windows 2000 Setup expands this to the computer name of the system where the upgrade check is being performed. For example:

```
SaveReportTo=\\server\share\%computername%.txt
```

When Windows 2000 Setup is run on a system with the computer name “Caroline”, this generates a report file whose fully qualified filename is

```
\\server\share\caroline.txt
```

If the answer file is not provided, Windows 2000 Setup saves the compatibility report to a file called Upgrade.txt in the Windows directory of the local hard disk. Note that Setup can only check for known incompatibilities. You should therefore test all of your programs to determine if they are compatible with Windows 2000.

Checking Windows NT Systems

When executed under Windows NT, Windows 2000 Setup does not give the option to save the report file centrally. The report is saved to Winnt32.log in the Windows directory. The file can be stored centrally using the following commands in a batch file. Note the switch used, CheckUpgradeOnlyQ (rather than CheckUpgradeOnly). This causes Winnt32 to generate the log file without displaying the Compatibility Check wizard.

```
winnt32.exe /CheckUpgradeOnlyQ
copy %windir%\winnt32.log \\server\share\%computername%.txt
del %windir%\winnt32.log
```

Executing the Winnt32 Command Line for Audit

The audit is fairly quick on Windows NT systems but may take some time on

Windows 9x systems. For this reason, you may choose to execute the check on Windows NT systems in a user logon script. However, you shouldn't execute the audit more than once; therefore, the script must contain some logic to enforce the run once requirement. For example, you might save a dummy file to the central location to indicate that the audit had been completed. The script would then test for the existence of this dummy file before executing Winnt32.

Alternatively, you can distribute the command line and Unattend.txt file with Microsoft Systems Management Server. Doing so enables you to exert more control over how and when the compatibility checking is conducted.

Note that when Windows 2000 Setup is started in a report-only mode, the user cannot continue with the upgrade or installation.

Windows 2000 Compatibility Lists

Ideally, the hardware and software you plan to use with Windows 2000 should be on the Windows 2000 compatibility lists. These can be found at http://www.microsoft.com/windows/compatible/default.asp and http://www.microsoft.com/hwtest/hcl

If you are using hardware or software that does not appear in these lists, you should contact the vendor to determine the availability of Windows 2000–compatible versions or updates. You may also find that the hardware or software is compatible but has simply not been tested. Confirm this with your own testing and with the assistance of the vendor.

If you plan to purchase new hardware or software, you should ensure that it is compatible to get the most reliable and easy-to-use Windows 2000 experience. You should consider making this one of the criteria that your purchasing department uses when making buying decisions.

Pre-upgrade Checks

Before upgrading to Windows 2000, you should have or know the following information:

- Know what hardware and software your organization uses. You can obtain this information by auditing your users' desktops or through a compatibility check.
- Determine how you are going to upgrade or replace incompatible hardware and software.
- Know the minimum hardware requirements for Windows 2000. These are detailed in the release notes on the Windows 2000 CD and can be found at http://ntbeta.microsoft.com/support/updates.asp
- Map the topography of your network. The geography of your sites and the available bandwidth between them is important when choosing to do push or on-demand installations. You should also note the number of users at each site. Remember to include remote users, both mobile users with laptop computers, and home-based users.

- Note how many servers you have at each site and determine how much available storage you have.
- Know how the timing of upgrades is determined and whether the central IT functions or local management controls this.

Also, be aware of the following issues (which are commonly overlooked):

- Physical access to computers including power-on passwords set in CMOS.
- Planned outages and disruptions.
- National holidays and religious festivals that can impact international deployments.
- Scheduled reporting periods; for example, end-of-year (both calendar and fiscal) and end-of-month reporting. These often vary between departments and political divisions of organizations.
- Staff vacations. In general, it is not ideal to upgrade a user's computer when that user is on vacation.
- Applications deployed at the departmental or geographic level. You will almost certainly find applications that are not part of the core suite. A thorough audit will help to identify these.

You should be sure to read all of the release notes and Readme files on the Windows 2000 CD. Also, ensure that you have the Windows 2000 Resource Kit. The Resource Kit contains many new and updated tools and utilities that can help you during the Setup and customization phases of installing Windows 2000.

PLANNING THE UPGRADE

Microsoft Consulting Services (MCS) consultants are often asked questions such as "What is it that organizations do that leads to a successful upgrade?" While this guide does not go into the detail of the planning process, it does provide the following quick suggestions. If you consider these points and manage the issues they address, your upgrade project has a better chance of running smoothly.

Microsoft Solutions Framework

The Microsoft Solutions Framework (MSF) provides models for project team formation and project planning. It is important to note that MSF is not a methodology. As its name implies, it is a framework and is based upon the best practices of Microsoft's development groups and MCS. MSF deployment planning is based on a milestone-driven approach with flat teams of peers. There is little hierarchy in the MSF team model.

More information on MSF can be found at http://www.microsoft.com/SolutionsFramework/AboutMSF.htm

Microsoft Consulting Services can assist you in implementing an MSF based approach.

Evaluations and Pilots

You should perform a thorough lab-based evaluation of Windows 2000. Assemble a representative sample of the PC hardware in use at your site in a test lab. Comprehensive testing on each hardware platform, combined with testing of application installation and operation, can greatly increase both the confidence of the project teams and that of the business decision makers, and ultimately leads to a higher quality deployment. Remember though that Microsoft has invested a great deal of time and money testing industry-wide hardware and software. Supplement this testing with your own.

Execute limited scale pilots. The primary purpose of pilot projects should not be to test Windows 2000. Rather, you should carry out early pilots, probably of no more than 50 users, to provide feedback to the project team. This feedback is used to determine the features that you need to enable or disable within Windows 2000. This is particularly relevant if you upgrade from Windows 9x where features such as domain-based machine accounts, local security, and file system security have not previously been addressed. The user population chosen for pilots should represent a cross-section of your business, both in terms of job function and IT proficiency.

Indeed, so important is this feedback that you need not necessarily test the upgrade mechanism. You may choose to upgrade pilot systems manually from CD or from a network installation point. When you have made all of the necessary design decisions, use a final pilot to test the upgrade mechanism.

The pilot results should be carefully compiled. You can use the results data to estimate upgrade times, the number of concurrent upgrades you can sustain, and peak loads on the user support functions.

Desktop Maintenance and Change Control

Plan and test how you are going to roll out upgrades, both to Windows 2000 and to applications. The Microsoft Windows Installer Service is new to Windows 2000 and can be an effective part of this strategy. Further information can be found at http://www.microsoft.com/windows/professional/technical/whitepapers

Additional information on change and configuration management is available at http://www.microsoft.com/windows/server/Technical/management/default.asp

Note that Microsoft Systems Management Server is ideal for maintaining networks of Windows 2000 computers.

Management Buy-In and Involvement

Department heads, line managers, and support functions should be involved at all stages of the process. Advertise your project and send regular updates on progress to all areas of the business. Early visibility of your project encourages useful feedback, often on issues you may not have considered. In addition, if managers are involved in the process, they are more likely to see the project in a positive light and assist you whenever they are able.

Make sure that users have training available and that support personnel are adequately trained and able to support the upgrade, including the pilots.

Microsoft Consulting Services and Microsoft Solutions Providers

It is almost certain that MCS and Microsoft Solutions Providers have seen the issues you are faced with. Make use of their services to reduce risk and cost to your project. Details of MCS and Microsoft Solutions Providers can be found at http://www.microsoft.com/enterprise/support/support/consult/consult_home.htm

http://www.microsoft.com/enterprise/support/support/partner/partner_home.htm

Involve MCS at the early stages of your decision making process. They will be able to make you aware of issues particular to your environment, and can help you to make the correct decisions around deployment. MCS is of less value when engaged in the later stages of a project to troubleshoot issues and when proposed changes to designs are harder and more expensive to implement. Microsoft Solutions Providers are ideally placed to help you with the longer-term implementation of your plan and have experience and knowledge that you can use effectively.

Plan, Plan, and Plan Again

In summary, any investment you make in planning will pay back many times over. Troublesome upgrades and deployments are nearly always a result of insufficient planning.

UPGRADE TOOLS AND MECHANISMS

The OEM directory and other customization features are mentioned in the following chapters. These features are described in detail in the section, “Customizing the Windows 2000 Installation,” in this document.

Either Winnt.exe or Winnt32.exe starts Windows 2000 Setup. (Winnt.exe starts Setup under 16-bit Microsoft operating systems only, and is not considered further in this guide.)

WINNT32

The command line options for Winnt32.exe that are most commonly used during an upgrade are described below. For a complete list of options, type

```
winnt32 /?
```

at a command prompt.

The purpose of using answer scripts and the Winnt32.exe command line options is to fully automate the upgrade process. There should be no user input. Remember that Winnt32.exe executed from a supported Windows 9x or Windows NT installation can perform an upgrade, retaining installed programs and user settings and preferences.

```
winnt32 [/s:sourcepath][/unattend[:answer_file]][/m:folder_name]
Parameters
/m:folder_name
```

Specifies that Setup copies replacement files from an alternative location. Instructs Setup to look in the alternative location first, and if files are present, to use them rather than the files from the default location.

```
/s:sourcepath
```

Specifies the location of the Windows 2000 files. To copy files from multiple servers simultaneously, specify multiple **/s** sources. If you use multiple **/s** switches, the first specified server must be available or Setup will fail.

```
/unattend
```

Upgrades your previous operating system in unattended Setup mode. All user settings are taken from the previous installation, so no user intervention is required during Setup.

Using the **/unattend** switch to automate Setup affirms that you have read and accepted the End User License Agreement (EULA) for Windows 2000. Before using this switch to install Windows 2000 on behalf of an organization other than your own, you must confirm that the user (whether an individual or a single entity) has received, read, and accepted the terms of the Windows 2000 EULA.

```
/unattend:[answer_file]
```

Performs an installation in unattended Setup mode. The answer file provides your custom specifications to Setup.

The unattended answer file can provide all the information needed to automate

Windows 2000 Setup. All existing user settings and preferences, for example mapped drives, desktop preferences and printers, are preserved.

It is beyond the scope of this guide to include a complete reference to the unattended answer file (the default file is Unattend.txt). Details can be found in the Windows 2000 Resource Kit and in Unattend.doc on the Windows 2000 CD.

Note Some sections are not relevant during an upgrade, and are therefore ignored.

Preparing the Distribution Point

This process is similar to that used in deploying previous versions of Windows NT.

To create a distribution point

1. Create a folder on a server.
2. Share this folder across the network. Remove create and write permissions for default users.
3. Create a user group for those people who will customize the distribution point, and add permissions to allow this group to customize the distribution point.
4. Copy the contents of the i386 directory to the shared folder.
5. Use Setup Manager to create the unattended answer file. The default name for this file is Unattend.txt.

Note All drivers included with the Windows 2000 CD are now contained in a single file, Driver.cab. This file is installed to %windir%\driver cache\i386 and the files within it are listed in Drvindex.inf. This was been done for several reasons, including:

- Installing new devices on laptops does not require access to the CD or the original network installation point.
- This reduces the disk footprint on distribution shares with large cluster sizes. Thousands of files have been replaced by one .cab file.
- This reduces installation time across a network. It is far more efficient to copy one large file than many small files.
- Most printer drivers use the Windows 2000 core printer drivers. If a new printer driver is installed, there is not requirement to have access to the Windows 2000 CD or the original network installation point.

To execute the unattended upgrade, connect to the shared folder and execute the following command line:

```
Winnt32.exe /s:folder_name /unattend:folder_name\answer_file
```

where *folder_name* is the name of the shared folder containing the distribution point and *answer_file* is the name of the answer file you created with Setup Manager.

For example, the following commands connect to a distribution folder shared as “i386” on a server named “WIN2000” and execute the unattended Setup. Note the use of colons following each command line switch.

```
net use x: \\WIN2000\i386
x:\winnt32 /s:x:\ /unattend:x:\unattend.txt
```

You may experience problems executing Winnt32 from a Windows 9x system when the distribution folder share point is mapped to the root of a drive letter. Setup may stop with an error message indicating that it could not access the Windows 2000 source files. To ensure that this does not occur, create the distribution point in a subfolder of the shared folder and specify this path in the commands. In the next example, the shared folder is named INSTALLS and the subfolder containing the distribution point is named i386.

```
net use x: \\WIN2000\INSTALLS
x:\i386\winnt32 /s:x:\i386 /unattend:x:\i386\unattend.txt.
```

By using the using OemFilesPath and OemPnPDriversPath keys in the [Unattended] section of the answer file, you can create a read-only distribution share point. You can then use this to safeguard the installation, ensuring that the distribution point cannot be accidentally modified. Specific access permissions to the OEM files and Plug and Play drivers can then be granted to individuals and user groups who require access.

Note that you can specify both the Win9xUpgrade and NtUpgrade keys in the answer file. Setup uses only the appropriate key. The presence of the [Win9xUpg] section is also ignored when you perform an upgrade from Windows NT. This means that you can use a single answer file for both upgrades.

UPGRADING WINDOWS 9X SYSTEMS

You can use Windows 2000 to upgrade all released versions of Windows 95, Windows 95 OSR2, and Windows 98.

Using an Answer File

The unattended answer file can be used to control how the Windows 9x upgrade is performed. If you are upgrading a Windows 9x system and you know that all of the hardware and software is supported under Windows 2000, you may perform an unattended upgrade simply by using the following command line:

```
winnt32.exe /unattend /s:source files
```

where *source_files* is the location of the distribution share point. Note that an answer file is not required in this case.

If you need to exercise more control over the upgrade process, for example you might want to process custom upgrade packs or additional Plug and Play device drivers, then you can use an answer file to control the upgrade.

The most important answer file sections and keys used during upgrades are as follows. (Refer to the Windows 2000 Resource Kit for details of all available options.) Note that there are significant changes from previous versions of Windows NT. Use of the OEM structure is now supported during upgrades (along with the processing of Cmdlines.txt) as is the ability to make additional Plug and Play device drivers available to Setup. The following sections and keys of the unattended Setup answer file can be used to take advantage of these features.

[Unattended]

- ***OemPreinstall***. Use this key to install additional software or customize the upgrade process. Details of the OemPreinstall mechanisms are provided later in this guide. The OEM structure and Cmdlines.txt file are both processed during an upgrade.
- ***Win9xUpgrade***. This key is used in conjunction with the Win9xUpg section of the answer file.
- ***OemFilesPath***. This key specifies the path to the OEM folder (containing OEM files) if it does not exist under the i386 folder of the distribution share point. The path can be a UNC name.
- ***OemPnpDriversPath***. Specifies the path to folders that contain Plug and Play drivers that do not ship on the Windows 2000 CD. The folders must contain all files necessary to install the particular devices—drivers, catalog, and .inf.

  For example, if you have a folder called *drivers* with subfolders called *audio* and *net*, you will specify OemPnPDriversPath = "*drivers\audio;drivers\net*" in the answer file. Setup will add %systemdrive% to each of the folder names and add those paths to the Plug and Play device search path.

  When using this parameter, ensure that the folders are available during GUI Mode Setup; you can use the OEM\$1 directory structure mechanism for this.

[Win9xUpg]

- ***MigrationDLLs***. This key specifies the location of upgrade packs that Setup needs to copy and process during an upgrade to Windows 2000. If multiple paths are specified, commas must separate the paths. Setup will search each of these paths (including its subfolders) for upgrade packs. Multiple upgrade packs can be located at a single path, but each upgrade pack must exist in its own subfolder of that single path. Do not put more than one upgrade pack in a single folder.

  An upgrade pack consists of a migration DLL (Migrate.dll) and any additional files that may be required to properly upgrade a particular software component from Windows 9x to Windows 2000.

  Corporate developers can use this mechanism to provide in-house upgrade packs for custom applications. Migration DLLs are standard Win32 DLLs. Windows 2000 Setup calls these DLLs during the upgrade process using standard published entry points. The migration DLL may then modify the system registry, install, move or remove files, recreate shortcuts to an applications components and carry out any other processing that is required to upgrade an application to work with Windows 2000. Full details for developers can be found at: http://msdn.microsoft.com/developer/windows2000/default.asp

  To add custom upgrade packs, install the migration DLL and any associated files it requires into a subfolder of a path referenced in the MigrationDLLs key. You do not need to supply any additional information in the answer file. Windows 2000 Setup will automatically search for and then call the migration DLL.

  Examples of upgrade packs can be found on the Windows 2000 CD in the i386\Win9xUpg folder.

  A single migration DLL can support multiple languages. You do not need to develop separate DLLs for each language upgrade you need to perform. Migration DLLs can be written so that they are called by Setup once for each user (based on the existence of user profiles). This enables upgrade packs to process per-user settings for all users of the computer.

- ***UserPassword***. This key informs Setup of the passwords to create for specific local accounts. Because Setup cannot migrate the Windows passwords of users when upgrading a system, it must create passwords for non-domain accounts during the migration process. With this key, an administrator can predetermine what those passwords will be for specific users.

  There are some security concerns associated with using this key because the password is stored as plain text within the answer file. However, after the upgrade is completed, all the password keys are deleted from the copy of the answer file left on the computer. The original copy of the answer file you started Setup with is not modified.

If a local account needs to be created for a user who does not have a UserPassword entry and no DefaultPassword is specified, Setup will create a random password. After the first reboot, Setup displays a dialog that asks you to pick a single password for both the administrator account and any other local user accounts it created with random passwords. The same password applies to everyone until the administrator logs on and changes the password(s).

- ***DefaultPassword***. This key provides a default password for all local accounts created during a migration process. Because Setup cannot migrate the Windows passwords of users when upgrading a system, it must create passwords for non-domain accounts during the migration process. When Setup needs to assign one of these passwords, it will first check to see if there is a UserPassword (see above) entry for that user. If not, it will use the value of this key if specified.

  There are some security concerns associated with using this key because the password is stored as plain text within the answer file. However, after the upgrade is completed, all the password keys are deleted from the copy of the answer file left on the computer. The original copy of the answer file you started Setup with is not modified.

  If a local account needs to be created for a user who does not have a UserPassword entry and no DefaultPassword is specified, Setup will create a random password. After the first reboot, Setup displays a dialog that asks you to pick a single password for both the administrator account and any other local user accounts it created with random passwords. The same password applies to everyone until the administrator logs on and changes the password(s).

- ***ForcePasswordChange***. This key informs Setup to automatically require a password change on all local accounts it creates during the migration process. When a user first logs on using one of these accounts, the user will be informed that the current password has expired and the user will be forced to select a new password before logging on. Note that the default setting for this option is Yes.

- ***MigrateUsersAsAdmin***. This key informs Setup to add all accounts that it creates during migration to the Local Administrators group, giving those users full control over the computer. Note that the default setting for this option is Yes.

- ***MigrateUsersAsPowerUsers***. This key informs Setup to add all accounts that it creates during migration to the Power Users group, giving those users full control over the computer.

- ***MigrateDefaultUser***. This key informs Setup to migrate the default Windows 9x user account settings to the default Windows 2000 user account. Note that the default setting for this option is No.

- ***UseLocalAccountOnError***. This key directs Setup to create a local account if a network account cannot be automatically determined or resolved. This is only

valid on computers with the Microsoft Networking Client software installed. Note that the default setting for this option is No.

However, Windows 9x only keeps the domain of the last logged user in its registry. It does not keep the domains of other users who may have logged on to the computer. Therefore, Windows 2000 Setup searches all trusted domains on the network by default and automatically uses a domain account when an exact match is found.

If a user is not found on any trusted domain or if the user account is found on two or more domains on the network, a dialog box is presented to the person performing the upgrade to resolve the conflict. This dialog box is also presented if network errors occur.

Specifying UseLocalAccountOnError=Yes in the answer file will ensure a complete unattended installation. This will cause Setup to create a local account whenever a network account cannot be automatically resolved.

Having a local account implies that the user may not have his or her original network privileges. In addition, if a computer cannot be added to the computer domain during installation of the network on Windows 2000, all user accounts will become local accounts.

- ***UserDomain***. This key specifies the user domain for a user. Multiple UserDomain lines can be used to specify different domains for different users. When specified, this key prevents Setup from searching all trusted domains on the network for a matching user account. (The search process can be time-consuming if there are a large number of trusted domains on the network.)

  If the account is not found in the specified domain (either because the account does not exist or the domain is not accessible), a dialog box is presented, and the user must resolve the account unless the UseLocalAccountOnError key is set to Yes.

Additional Considerations when Upgrading from Windows 9x

Installation Directory

It is not possible to specify the installation directory during the upgrade. Windows 2000 will be installed in the same directory where Windows 9x was installed. For example, if Windows 95 is installed in c:\windows and this installation is upgraded to Windows 2000, Windows 2000 will also be installed in c:\windows. The TargetPath key in the [Unattended] section of the answer file is ignored.

Machine Accounts

Unlike Windows 9x, Windows 2000 computers that are to participate in a Microsoft network domain require a machine account on the domain in which they are to participate. This is so that the domain controller servers can identify the computer and grant access to domain resources. Without this machine account, users have to logon to the local workstation domain or workgroup and then explicitly authenticate

themselves when connecting to domain resources. Machine accounts are not required for Windows 2000 computers to access non-Microsoft network servers.

There are three ways to create machine accounts for Windows 9x computers being upgraded to Windows 2000:

- Create the machine account on the domain before the upgrade is performed.
- Create the machine account during the winnt32 phase of Setup.
- Create the machine account after the upgrade has been performed.

Each of these three methods can be done manually or automatically.

Creating Machine Accounts Before Performing the Upgrade

Note If you are not running Microsoft Windows NT servers, desktop computers do not need machine accounts.

Domain administrators can create machine accounts manually through the Server Manager administration tool on a Windows NT or Windows 2000 domain server. However, to do this manually would be an extremely time consuming process for anything more than a few computers. It is therefore preferable to automate the process of creating machine accounts.

To automate the machine accounts, you need a listing or database of the machine names and the domains in which the machine accounts will be created. There are several ways in which this machine account creation can be automated. The Active Directory Services Interface (ADSI) offers a flexible solution.

Important Despite its name, ADSI supports more directories than the Active Directory. For example, it supports NTDS, NDS, and LDAP–compliant stores. You can therefore use ADSI to interface to the directory in a Windows NT domain.

ADSI abstracts the underlying Directory Services by using a standard programming model that can be used from Windows Scripting Host, Microsoft Visual Basic, and other development tools that support automation of COM objects. For more information on ADSI, refer to:
http://www.microsoft.com/NTServer/nts/exec/overview/ADSIfaqs.asp

Using ADSI, a script or utility program can be written to extract the computer names and domain names from the database and automatically create the machine accounts. The database of computer names and domains can be compiled as part of the audit, for example using SMS.

Note Creating machine accounts before upgrading is the preferred method.

Creating Machine Accounts During the Upgrade

There are three main ways in which a machine account can be created during the upgrade.

- By specifying the domain to join in the unattended answer file
- By allowing the user to provide the necessary information

- By using the Netdom Resource Kit utility in a run-once command line

The disadvantage of all three of these methods is that a user name and password for an account that has permissions to create machine accounts is required. In the case of using the answer file or automating the Netdom utility, these names and passwords could be stored centrally. In any case, you should consider the security implications of storing these details in a file or allowing end-users to create their own machine accounts.

If you supply a domain account and password in the unattended answer file, Setup will automatically remove these from the local copies of the answer file left on the computer after the upgrade. It may therefore be acceptable to create a special account for your upgrades and remove this account once the upgrades are completed.

Creating the Machine Account after Setup has Completed

If a machine account is not created before or during the upgrade, the computer will not be able to join a domain. Instead, it will join a workgroup. Joining a workgroup can be controlled using the unattended answer file. A domain does not authenticate users logging onto a Windows 2000 computer in a workgroup. Rather, they are authenticated by the computer's local or workstation domain using a user account local to the computer. To gain access to domain resources, users will have to explicitly authenticate themselves to the domain using their domain account. One of the great advantages of the domain model is that a single logon can be used to authenticate users to the local computer, the logon domain (the one containing the users domain account) and any domains that trust the logon domain. In this way, users can be granted seamless access to multiple resources across several domains.

A computer can join a domain (and create a machine account) once Setup has completed.

To join a domain

1. Right-click the My Computer icon.
2. Select **Properties** from the context menu.
3. Click the **Network Identification** tab, and then click **Change**.

The Network Identification Wizard will guide you through the process of joining a domain. Alternatively the Netdom utility could be used from a command line or script. Note that the user (or script) will need to have the user name and password for an account with the appropriate permissions.

It is also possible to create utilities that make a request to create a machine account to a central server computer without having to provide a user name and password. For example, within Microsoft there are web-based utilities on the Microsoft intranet that allow users to maintain their own machine accounts. The server application is granted the appropriate permission to do this, removing the need for the client computer (or user) to know the account details. However, this does mean that

anyone with physical access to a computer on the network and a domain user account can create a machine account and thus connect an unauthorized computer.

For these reasons, it is preferable to create the machine account before the upgrade. Doing so will ensure that the upgrade is fully unattended and that user profiles are correctly migrated.

User Accounts and Profiles

During the upgrade, any user profiles on the Windows 9x computer are migrated to Windows 2000 profiles. Windows 2000 maintains two types of user accounts: domain accounts and local accounts. In addition, accounts can be members of groups: either global groups accessible to all computers in the domain, or local groups accessible only to the computer on which they are defined.

If a domain controller of the domain containing the user's domain account cannot be contacted during the upgrade, either because the domain controller is not available or there is a problem with the networking on the computer, Windows 2000 Setup cannot create a profile for that domain account. If this happens, the Windows 9x profile is migrated to a local user account profile. For this reason, you should ensure that the computer is connected to the domain and that the domain controllers are available during the upgrade.

Once Setup is complete, it is possible to copy this local account profile to the user's domain account profile but desktop settings and other preferences might not be preserved.

To copy a profile

1. Right-click the My Computer icon.
2. Select **Properties** from the context menu.
3. Click the **User Profiles** tab.
4. Select the profile you want to copy, and click **Copy To**.

Windows 2000 does not allow you to use a user account name that is the same as one of the built-in accounts or groups. For example, a Windows 9x user account named "Administrators" is not allowed under Windows 2000. This account will therefore be renamed during Setup. During an unattended upgrade, Windows 2000 will generate compatible names. The names to be used are written to the compatibility report.

Make sure that the computer is connected to the network and that the domain controller is available during the upgrade. When upgrading laptops, it is best to schedule this when the user is able to connect to the corporate network.

Choosing the File System

You can determine how Windows 2000 Setup treats the file system by using the FileSystem key in the Unattended section of the answer file. Windows 2000 recognizes FAT16, FAT32 and NTFS. Using the FileSystem key, you can either leave the file system alone or convert it to NTFS. As Windows 9x does not support NTFS, the choice is whether to leave the file system as FAT or convert to NTFS. There are many advantages to using NTFS. Amongst these are:

- Increased robustness—NTFS is a transactional file system and can automatically recover from many errors.
- Increased security—access to files can be secured and files and folders can be encrypted.

- Support for large media.
- Faster access.

Microsoft recommends that you format all Windows 2000 partitions with NTFS, with the exception of the system partition of an Alpha architecture computer and dual boot configurations.

Please refer to the Windows 2000 Resource Kit or Release Notes for a more thorough comparison of these file systems.

UPGRADING WINDOWS NT SYSTEMS

Windows 2000 can be used to upgrade Windows NT 3.51 and 4.0 servers and workstations.

Using an Answer File

The unattended answer file can be used to control how the Windows NT upgrade is performed. If you are upgrading a Windows NT system and you know that all of the hardware and software is supported under Windows 2000, you may perform an unattended upgrade simply by using the following command line:

```
winnt32.exe /unattend /s:source files
```

where *source_files* is the location of the distribution share point. Note that an answer file is not required in this case.

If you need to exercise more control over the upgrade process—for example, if you want to process additional Plug and Play device drivers—you can use an answer file to control the upgrade.

The most important answer file sections and keys used during an upgrade are explained below. Refer to the Windows 2000 Resource Kit or the Unattend.doc file on the Windows 2000 CD for details of all available options. Note that there are significant changes from previous versions of Windows NT.

- ***OemPreinstall***. Use this key to install additional software or customize the upgrade process. Details of the OemPreinstall mechanisms are provided later in this guide. The OEM structure and cmdlines.txt file are both processed during an upgrade.
- ***NTUpgrade***. This key determines whether Setup upgrades an existing Windows NT system or performs a clean install into a new directory.
- ***OemFilesPath***. This key specifies the path to the OEM folder (containing OEM files) if it does not exist under the i386 folder of the distribution share point. The path can be a UNC name.
- ***OemPnpDriversPath***. Specifies the path to folders that contain Plug and Play drivers that do not ship on the Windows 2000 CD. The folders must contain all files necessary to install the particular devices—drivers, catalog, and .inf.

  For example, if you have a folder called *drivers* with subfolders called *audio* and *net*, you will specify OemPnPDriversPath = "*drivers\audio;drivers\net*" in the answer file. Setup will add %systemdrive% to each of the folder names and add those paths to the Plug and Play device search path.

  When using this parameter, ensure that the folders are available during GUI Mode Setup. You can use the OEM\$1 directory structure mechanism for this.

Using Dosnet.inf to Detect Incompatibilities

Windows 2000 Setup uses Dosnet.inf to determine which services, drivers or applications are incompatible with Windows 2000. When Winnt32 is executed, Dosnet.inf is loaded and parsed, and Setup determines what actions to take based upon the entries in this file.

Layout of Dosnet.inf

Dosnet.inf is similar to any other .inf file. It contains sections denoted by angle brackets and keys within those sections to control Setup processes. Among others, the two sections used to control detection of incompatibilities are:

```
[ServicesToDisable]
[ServicesToStopInstallation]
```

Winnt32 uses entries in these sections to search for incompatibilities. There are four ways to detect and report incompatibilities:

- Registry entries
- Existence of files
- Existence of a Windows NT Service (the preferred method)

The action Winnt32 takes depends on the entry and the section where it is placed. If the entry is in ServicesToDisable and the entry refers to a Windows NT Service, the user is warned (if the upgrade is not running in an unattended mode) and the service is disabled. If the entry is not a service, the user is warned that there is a possible incompatibility, but no further action is taken.

If the entry is in ServicesToStopInstallation, the user is warned and the upgrade is halted. This occurs for all four types of entry. Setup cannot continue until the incompatibility has been resolved, usually by removing the component or applying an upgrade.

Note that with the exception of disabling services, no change is made to the system; for example, files are not deleted, registry keys are not updated, and so on.

Entries Used to Detect Incompatibilities

To detect the presence of a particular registry key or value, add the following entry to either section.

```
r, key_name, value_name, expected_value, html_file, text_file,
%description%
f, file_name, version, html_file, text_file, %description%
s, service_name, html_file, text_file, %description%
```

where:

- *key_name* is the registry key to search for. Note that the hive name should be the full name rather than the abbreviation:
  Correct. HKEY_LOCAL_MACHINE
  Incorrect. HKLM
- *value_name* is the value name to search for under the key specified in *key_name*.
- *expected_value* is the registry value to search for. If it is found, then the

appropriate action is taken according to which section the entry appears in.

- *html_file* is the name of a .htm file that will be displayed when the entry is found.
- *text_file* is the name of a .txt file that will be displayed when the entry is found.
- *%description%* is the "friendly" name of the incompatibility displayed to the user. As with all .infs, this friendly name is expanded with a corresponding entry in the [Strings] section of Dosnet.inf.

  [Strings]

  Description = "A description of the incompatibility"
- *file_name* is a fully qualified file name of a file to detect.
- *version* is the file version of *file_name* to detect—for example, 5.1. If this version is not detected, no action is taken. You can use multiple file entries to detect multiple versions.
- *service_name* is the name of a Windows NT service as it appears in the registry. It is the name as shown in the Services Control Panel utility. If the service is to be disabled, Winnt32 writes an entry to Winnt.sif for text mode Setup to disable the service. Note that most device drivers run as services.

The html_file and text_file are used to give more information to the user if they click on the incompatibility, and then click **Details** on the wizard page. The HTML file can be used to point the user to updates to resolve the incompatibility. To display the .htm file, you must have Internet Explorer 3.0 or greater installed. Otherwise, the text file is displayed instead. These parameters must be specified, even if they point to dummy files, or the entry cannot be processed.

The default location for files is rooted at the source files directory. For example, the default Dosnet.inf contains information on compatibility issues in the COMPDATA subfolder of the i386 folder. To specify a file in the COMPDATA subfolder, use compdata\myfile.htm. You can also use %systemroot% and %windir% when locating files and services.

Dosnet.inf is loaded and parsed when performing an upgrade check in the report only mode. You can therefore modify this file to include your own known incompatibilities in advance of performing the audit.

For more information on detecting incompatibilities, please refer to:
http://www.microsoft.com/HWDev/desinit/NTSetup.htm

Choosing the File System

When Windows 2000 is installed on an NTFS partition, the partition will be automatically converted to NTFS 5. You can choose to upgrade FAT partitions or leave them as they are through the use of the FileSystem key in the [Unattended] section of the answer file.

HOW WINDOWS 2000 SETUP WORKS

As with previous releases of Windows NT, there are three distinct phases to Windows 2000 Setup:

- The Winnt32 phase
- The text mode phase
- The GUI mode phase

Each of these is described separately, below.

WINNT32 Phase

The Winnt32 phase loads Dosnet.inf, runs the compatibility check, and prepares the system for text mode Setup. If Winnt32 is being run with the /checkupgradeonly switch, Winnt32 exits after the compatibility check.

When Winnt is run to perform an installation from real mode, the user has the option of creating four boot floppies that are used to begin the installation. During an upgrade or unattended installation, this does not happen. Instead, Winnt32 copies these same files to a temporary folder called $win-nt$.~bt, on the installation hard disk. These files are used to boot the computer after the reboot at the end of the Winnt32 phase. If you need to add support for custom boot devices, you can do this by using the OEM structure and some required keys in the answer file. See the next section of this guide for details.

A Boot.ini file is created, and loads Bootsect.dat in nwin_nt.~bt after the reboot at the end of the Winnt32 phase. Bootsect.dat contains a bootstrap loader for the text mode phase of Setup.

Winnt32 can also copy the Windows 2000 source files to a temporary directory called win_nt.~ls, on the installation hard disk. If the upgrade is being performed from a CD ROM supported by Windows 2000, this copying does not take place (this saves time and temporary storage space on the hard disk). This directory is a copy of the i386 directory, and its subdirectories, from the distribution share point.

Windows 9x Upgrade Specific Actions

When upgrading Windows 9x systems, the Winnt32 phase of Setup does the following:

1. Prompts the user for migration DLLs and files to resolve Windows 2000 incompatibilities
2. Scans the Windows 9x system for installed applications and devices.
3. Notes any incompatibilities with Windows 2000.
4. Scans the Windows 9x installation for all user profiles, including the default user profile.
5. Presents a report to the user (if the upgrade is not running in an unattended mode) of device and application incompatibilities.

Windows NT Upgrade Specific Actions

When upgrading Windows NT systems, the Winnt32 phase of Setup does the following:

1. Loads and parses Dosnet.inf
2. Presents a report to the user (if the upgrade is not running in an unattended mode) of device and application incompatibilities.
3. Prepares Winnt.sif (a Setup translated form of the answer file) to disable any services that are listed in the ServicesToDisable section of Dosnet.inf.

Text Mode Phase

This phase begins after the first reboot following the Winnt32 phase. The text mode phase installs the Windows 2000 operating system kernel and prepares the computer for the GUI mode phase of Setup.

Windows 2000 is installed into the same directory as the operating system being upgraded and the computer is prepared to boot into the final, GUI mode phase of Setup.

The root directory now contains Ntldr, Ntdetect.com, TxtSetup.sif and Pagefile.sys. The Boot.ini is set to start the new Windows 2000 installation with a timeout of 0 seconds (to prevent the user from selecting an alternative boot system). This timeout will be modified to the default of 30 seconds during the GUI mode phase. The system is then rebooted.

GUI Mode Phase

GUI mode Setup detects devices and enumerates Plug and Play devices. A list of devices to be installed is compiled for use later in this phase. All .inf files are now pre-compiled to .pnf files, and the system then attempts to load and initialize previously detected devices. Services that will be running once Setup is complete are also started.

It is during this phase that Setup customizes Windows 2000 with the answer supplied by the answer file or typed in by the user during the upgrade. This includes things such as the user and keyboard locales, the regional settings, time zone, and so on. Note that most settings are retained from the previous installation when performing an upgrade. The goal is to make Windows 2000 closely match the previous installation following the upgrade.

Application migration is performed. Any migration DLLs provided during a Windows 9x upgrade are called and processed.

Windows 2000 networking is installed and the network is started. The computer joins the domain, if appropriate.

DLLs are registered and their corresponding registry keys are created. The system registry hives are created and saved into %windir%\system32\config. Windows 9x profiles are migrated. Windows 9x cabinet files are removed if present. All final file

copies are completed and the temporary directories are removed.

The system is now installed and the computer reboots, ready for the first logon.

CUSTOMIZING THE WINDOWS 2000 INSTALLATION

There are several mechanisms through which you can customize the Windows 2000 upgrade process.

OEM Directory Structure

If the [Unattended] section of the answer file has the key OemPreinstall = Yes, the OEM directory structure and the Cmdlines.txt file are processed. If the /unattend switch of Winnt32 is specified with no answer file, this structure will not be processed.

The OEM structure can be used to install additional software or to replace Windows 2000 components. To create a OEM structure you can use the Windows 2000 Setup Manager or create the folders and subfolders manually. The root of the OEM structure is at a subfolder named OEM of the i386 directory.

The complete structure with the use of each subfolder is illustrated in Figure 1, below. Note that you can use the OemFilesPath key in the [Unattended] section of the answer file to modify the location of the OEM structure.

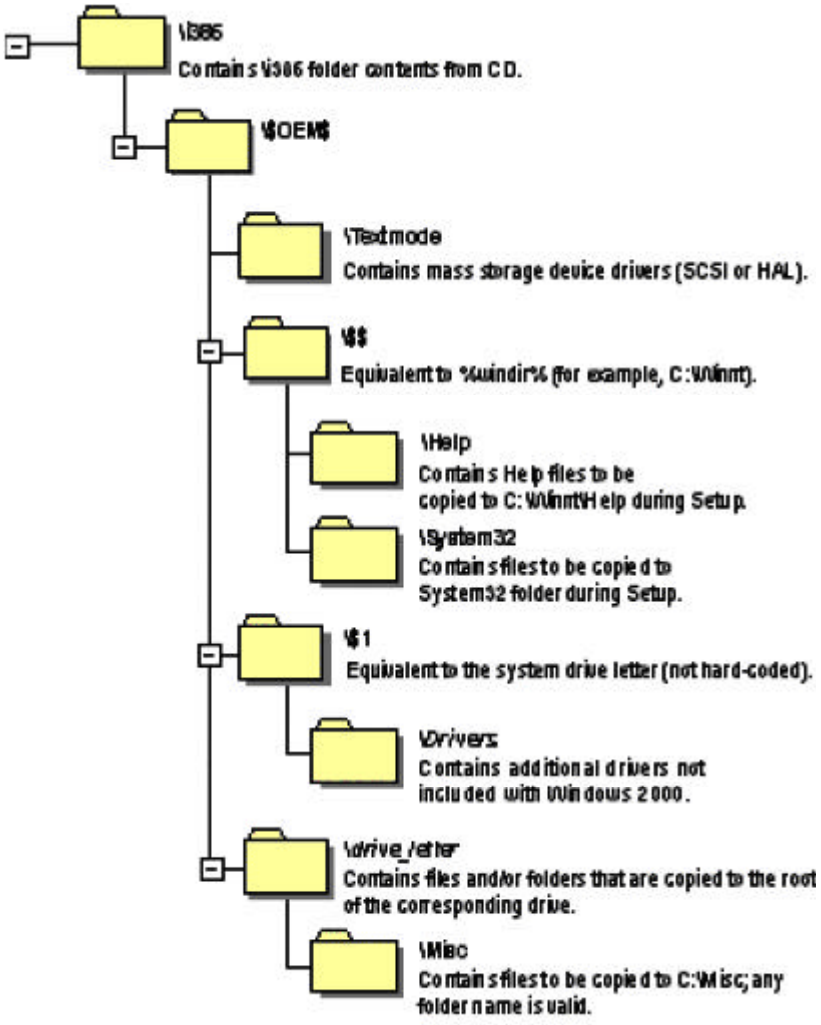

Figure 1. OEM Structure

OEM Folder

This folder, which you create in the distribution folder directly below the i386 folder, contains all the additional files required.

The OEM folder can include the optional file Cmdlines.txt, which contains a list of commands to be run during GUI-mode Setup. These commands can, for example, run a Windows 95 style .inf file, an application Setup command, Sysdiff.exe, or other executable file.

OEM\Textmode Folder

This folder contains the hardware-dependent files that Setup Loader and text mode Setup install on the target computer during text mode Setup. These files can include OEM hardware abstraction layers (HALs), SCSI device drivers and TxtSetup.oem, which directs the loading and installing of these components.

For x86-based computers, be sure to include the TxtSetup.oem file and all the files listed in it (HALs and drivers) in the [OEMBootFiles] section of Unattend.txt.

OEM\$$ Folder

This folder contains the system files (either new files or replacements for retail files) that are copied to the various subfolders when Windows 2000 is installed. The structure of this folder must match the structure of a standard Windows 2000 installation, where \$OEM$\$$ matches \%Windir%, \$OEM$\$$\System32 matches \%Windir%\System32, and so on. Each subfolder should contain the files that need to be copied to the corresponding system folder on the target computer.

OEM*Drive_letter* Folder

Each OEM*Drive_letter* folder contains a subfolder structure that is copied to the root of the corresponding drive in the target computer during text mode Setup. For example, files you put in an OEM\C folder are copied to the root of the C drive. You can also create subfolders in these folders. For example, OEM\D\Misc creates a Misc folder on drive D.

Text mode Setup can only copy files with short filenames in 8.3 format. Files that have to be renamed should be listed in $$Rename.txt.

OEM\$1 Folder

This folder is equivalent to the %systemdrive% environment variable. For example, if the operating system is installed on drive C, OEM\$1 points to drive C. The use of a variable makes it possible to rearrange drive letters without creating errors in applications that point to a hard-coded drive letter.

OEM\$1*Drivers* Folder

This folder replaces the \Display and \Net folders used in a Windows NT 4.0 distribution share point. It contains additional drivers not included with Windows 2000.

$$Rename.txt File

The $$Rename.txt file changes short file names to long file names during Windows 2000 Setup. This file lists all the files in a particular folder that need to be

renamed. Each folder containing short file names that should be renamed must contain its own $$Rename.txt file.

To use the $$Rename.txt file, put the file in a folder that contains files that need to be converted.

Please refer to the Windows 2000 Resource Kit for the syntax of and additional information about the $$Rename.txt file.

Adding Support for Plug and Play Devices

Plug and Play devices that are not mass storage devices and are not included on the Windows 2000 CD can be easily pre-installed by following the steps below. This method works for audio, video, modem, and printer devices.

To pre-install a Plug and Play device

1. Create a subfolder in the distribution folder for any special Plug and Play drivers and their .inf files. For example, you can create a folder called PnPDrvs:

   ```
   $OEM$\$1\PnPDrvs
   ```

2. Add the path to the list of Plug and Play search drives by adding the following line to the Unattend.txt file:

   ```
   OEMPnPDriversPath = "PnPDrvs"
   ```

3. If you have subfolders in the PnPDrvs folder, you must specify the path to each subfolder. The paths must be separated by semicolons. For example, if the PnpDrvs folder contains the subfolders “Net” and “Audio,” the answer file should contain the following line:

   ```
   OEMPnPDriversPath = "PnPDrvs\Net;PnPDrvs\Audio"
   ```

Using Cmdlines.txt to Install Optional Components

The Cmdlines.txt file contains the commands that GUI mode Setup executes when installing optional components, such as applications. For example, the commands specified in this file can run an .inf file using the Rundll32.exe command, run Sysdiff.exe, or perform some other action you need performed. If you plan to use Cmdlines.txt to install an application, be sure to place the application you are installing in the OEM subfolder of the distribution folder.

The syntax for Cmdlines.txt is as follows:

```
[Commands]
"command_1"
"command_2"
"command_3"

      .
      .
      .

"command_x"
```

where *command_1*, *command_2*, *command_3*, and so on refer to the commands you want to run (and the order that they should be run) when GUI-mode Setup calls Cmdlines.txt. Note that all commands must appear in quotation marks.

Limitations of and Usage Notes on Cmdlines.txt

When using Cmdlines.txt, you should be aware of the following:

- Cmdlines.txt runs as a service rather than as a logged on user with network capability. Therefore, any user-specific information is written to the default user registry, and all subsequently created users also get that information.
- Cmdlines.txt requires that the files necessary for an application or utility to run be placed in the distribution folders.
- When installing applications by using Cmdlines.txt, you should install the application either in quiet (unattended) mode or through Sysdiff. The user should not have to interact with or answer questions about the application.
- If you are applying a Sysdiff package, you should use the /m parameter with Sysdiff. For example, if you are logged on as Administrator when you install an application and you create a difference file, some data might be placed in the Profiles\Administrator folder. However, during GUI-mode Setup, this folder may not exist. The /m switch places the per-user data in the default user profile structure.

Using a Batch File to Install Applications from the Startup Group

You can use a batch file to install applications from the Startup group as described in the following procedure.

To use a batch file

1. Create a batch file containing lines similar to the following example:

```
Start /wait path\Setup
Start /wait Del C:\Winnt\Profiles\All Users\Startup\filename.lnk
     Exit
```

   where

   - *path* is the path to the executable file that starts the installation.
   - *Setup* is the name of the executable file that starts the installation.
   - *Filename.lnk* is the name of the shortcut to the batch file.

2. Copy the batch file to the distribution folder or another folder that can be accessed during Setup.
3. Create the *filename*.*lnk* (shortcut) file, in 8.3 format, pointing to the batch file in the distribution folder.
4. Copy the *filename*.*lnk* file from the source computer to the oem\$1\docume~1\alluse~1\startm~1\programs\startup folder.

When the computer is restarted and runs in GUI mode Setup, the application is installed, and then the *filename*.*lnk* file is deleted from the Startup group.

FOR MORE INFORMATION

For the latest information on Windows 2000, visit our World Wide Web site at http://www.microsoft.com/windows/, the Windows NT Server Forum on MSN™, and The Microsoft Network online service (GO WORD: MSNTS).